Rotation-invariant Texture Recognition

Javier A.Montoya-Zegarra[1,2], João P. Papa[2], Neucimar J. Leite[2], Ricardo d. S. Torres[2], and Alexandre X. Falcão[2]

Computer Engineering Department[1], Faculty of Engineering, San Pablo Catholic University,
Av. Salaverry 301, Vallecito, Arequipa, Peru
Institute of Computing[2], State University of Campinas,
Av. Albert Einstein 1216, Campinas, São Paulo, Brazil
{jmontoyaz,papa.joaopaulo}@gmail.com

Abstract. This paper proposes a new texture classification system, which is distinguished by: (1) a new rotation-invariant image descriptor based on Steerable Pyramid Decomposition, and (2) by a novel multi-class recognition method based on Optimum Path Forest. By combining the discriminating power of our image descriptor and classifier, our system uses small size feature vectors to characterize texture images without compromising overall classification rates. State-of-the-art recognition results are further presented on the Brodatz dataset. High classification rates demonstrate the superiority of the proposed method.

1 Introduction

In the last years, several image recognition systems have been proposed in the literature as a result of many research efforts [1, 2]. Although those approaches have achieved high classification rates, most of them have not been widely evaluated in texture image databases. Traditionally, texture images may be characterized by: (1) small inter-class variations, i.e, textures belonging to different classes may appear quite similar, especially in terms of their global patterns (coarseness, smoothness, etc.), and (2) the presence of image distortions such as rotations. In this sense, texture pattern recognition is a still open task. The next challenge in texture classification should be, therefore, to achieve rotation-invariant feature representations for *non-controlled* environments. To address some of these limitations, this work proposes a new texture classification method, which is characterized by: (1) a new texture image descriptor based on Steerable Pyramid Decomposition, which encodes the relevant texture information in small size feature vectors including rotation-invariant characterization, and (2) a novel multi-class object recognition method based on the Optimum Path Forest classifier [3].

Roughly speaking, a Steerable Pyramid is a method by which images are decomposed into a set of multi-scale, and multi-orientation image subbands, where the basis functions are directional derivative operators [4]. Our motivation in using Steerable Pyramids relies on that, unlike other image decomposition methods, the feature coefficients are less affected by image distortions. Furthermore, the Optimum Path Forest Classifier is a recent approach that handles non separable classes, without the necessity of using boosting procedures to increase its performance, resulting thus in a faster and more accurate classifier for object recognition. By combining the discriminating

power of our image descriptor and classifier, our system uses small size feature vectors to characterize texture images without compromising overall classification rates. In this way, texture classification applications, where data storage capacity is a limitation, are further facilitated.

The outline of this paper is as follows. In the next section, we briefly review the fundamentals of the Steerable Pyramid Decomposition. Section 3 describes how texture images are characterized to obtain rotation-invariant representations. Section 4 introduces the Optimum Path Forest classifier method. The experimental setup conducted in our study is presented in Section 5. In section 6, experimental results on several datasets are given and are used to demonstrate the recognition accuracy improvement of our approach. Comparisons with other texture feature representations and classifiers are further discussed. Finally, some conclusions are drawn in Section 7.

2 Steerable Pyramid Decomposition

The Steerable Pyramid Decomposition is a linear multi-resolution image decomposition method, by which an image is subdivided into a collection of subbands localized at different scales and orientations [4]. Using a high-, and low-pass filter (H_0, L_0) the input image is initially decomposed into two subbands: a high-, and a low-pass subband, respectively. Further, the low-pass subband is decomposed into K-oriented band-pass portions $B_0, \ldots, B_{K-1}$, and into a lowpass subband L_1. The decomposition is done recursively by subsampling the lower low-pass subband (L_S) by a factor of 2 along the rows and columns. Each recursive step captures different directional information at a given scale. Considering the polar-separability of the filters in the Fourier domain, the first low-, and high-pass filters, are defined as [5]:

$$L_0(r,\theta) = L\left(\frac{r}{2},\theta\right)/2 \qquad H_0(r,\theta) = H\left(\frac{r}{2},\theta\right) \tag{1}$$

where r, θ are the polar frequency coordinates. L, H are raised cosine low-, and high-pass transfer functions:

$$L(r,\theta) = \begin{cases} 2 & r \leq \frac{\pi}{4} \\ 2\cos\left(\frac{\pi}{2}\log_2\left(\frac{4r}{\pi}\right)\right) & \frac{\pi}{4} < r < \frac{\pi}{2} \\ 0 & r \geq \frac{\pi}{2} \end{cases} \tag{2}$$

$$B_k(r,\theta) = H(r)G_k(\theta), \qquad k \in [0, K-1] \tag{3}$$

$B_k(r,\theta)$ represents the K directional bandpass filters used in the iterative stages, with radial and angular parts, defined as:

$$H(r,\theta) = \begin{cases} 1 & r \geq \frac{\pi}{4} \\ \cos\left(\frac{\pi}{2}\log_2\left(\frac{2r}{\pi}\right)\right) & \frac{\pi}{4} < r < \frac{\pi}{2} \\ 0 & r \leq \frac{\pi}{2} \end{cases} \tag{4}$$

$$G_k(\theta) = \begin{cases} \alpha_K\left(\cos\left(\theta - \frac{\pi k}{K}\right)\right)^{K-1} & \left|\theta - \frac{\pi k}{K}\right| < \frac{\pi}{2} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $\alpha_k = 2^{(k-1)}\frac{(K-1)!}{\sqrt{K[2(K-1)]!}}$.

3 Texture feature representation

This section describes the proposed modification of Steerable Pyramid Decomposition to obtain rotation-invariant representations, used to characterize the texture images.

3.1 Texture representation

Roughly speaking, texture images can be seen as a set of basic repetitive primitives characterized by their spatial homogeneity [6]. By applying statistical measures, this information is extracted, and used to capture the relevant image content into feature vectors. More precisely, by considering the presence of homogeneous regions in texture images, we use the mean (μ_{mn}) and standard deviation (σ_{mn}) of the energy distribution of the filtered images (S_{mn}). Given an image $I(x, y)$, its Steerable Pyramid Decomposition is defined as:

$$S_{mn}(x, y) = \sum_{x_1} \sum_{y_1} I(x_1, y_1) B_{mn}(x - x_1, y - y_1) \tag{6}$$

where B_{mn} denotes the directional bandpass filters at stage $m = 0, 1, \ldots, S - 1$, and orientation $n = 0, 1, \ldots, K - 1$. The energy distribution ($E(m, n)$) of the filtered images at scale m, and at orientation n is defined as:

$$E(m, n) = \sum_{x} \sum_{y} |S_{mn}(x, y)| \tag{7}$$

Additionally, the mean (μ_{mn}) and standard deviation (σ_{mn}) of the energy distributions are found as follows:

$$\mu_{mn} = \frac{1}{MN} E_{mn}(x, y) \quad \sigma_{mn} = \sqrt{\frac{1}{MN} \sum_{x} \sum_{y} \left(|S_{mn}(x, y)| - \mu_{mn}\right)^2} \tag{8}$$

The corresponding feature vector ($\boldsymbol{f}$) is defined by using the mean and standard deviation as feature elements. It is denoted as:

$$\boldsymbol{f} = [\mu_{00}, \sigma_{00}, \mu_{01}, \sigma_{01}, \ldots, \mu_{S-1K-1}, \sigma_{S-1K-1}] \tag{9}$$

3.2 Rotation-invariant representation

Rotation-invariant representation is achieved by computing the dominant orientation of the texture images followed by feature alignment. The **dominant orientation** (DO) is defined as the orientation with the highest total energy across the different scales considered during image decomposition. It is computed by finding the highest accumulated energy for the K different orientations considered during image decomposition:

$$DO_i = max \left\{ E_0^{(R)}, E_1^{(R)}, \ldots, E_{K-1}^{(R)} \right\} \tag{10}$$

where i is the index where the dominant orientation is found, and:

$$E_n^{(R)} = \sum_{m=0}^{S-1} E(m,n), \quad n = 0, 1, \ldots, K-1. \tag{11}$$

Note that each $E_n^{(R)}$ covers a set of filtered images at different scales but at same orientation. Finally, rotation-invariance is obtained by shifting circularly feature elements within the same scales, so that first elements at each scale correspond to dominant orientations. This process is based on the assumption that to classify textures, they should be rotated so that their dominant directions are the same. Further, it has been proved that image rotation in spatial domain is equivalent to circular shift of feature vector elements [7].

4 Texture feature recognition

This section aims to present the new approach to pattern recognition called OPF (Optimum Path Forest), which has been demonstrated to be generally more efficient than Artificial Neural Networks and Support Vector Machines [3]. The OPF approach works by modeling the patterns as being nodes of a graph in the feature space, where every pair of nodes are connected by an arc (complete graph). This classifier creates a discrete optimal partition of the feature space such that any unknown sample can be classified according to this partition. This partition is an optimum path forest computed in $\Re^n$ by the image foresting transform (IFT) algorithm [8].

Let Z_1, Z_2, and Z_3 be training, evaluation, and test sets with $|Z_1|$, $|Z_2|$, and $|Z_3|$ samples such as feature vectors. Let $\lambda(s)$ be the function that assigns the correct label i, $i = 1, 2, \ldots, c$, from class i to any sample $s \in Z_1 \cup Z_2 \cup Z_3$. Z_1 and Z_2 are labeled sets used to the design of the classifier and the unseen set Z_3 is used to compute the final accuracy of the classifier. Let $S \subset Z_1$ be a set of prototypes of all classes (i.e., key samples that best represent the classes). Let v be an algorithm which extracts n attributes (texture properties) from any sample $s \in Z_1 \cup Z_2 \cup Z_3$ and returns a vector $\boldsymbol{v}(s) \in \Re^n$. The distance $d(s,t)$ between two samples, s and t, is the one between their feature vectors $\boldsymbol{v}(s)$ and $\boldsymbol{v}(t)$ (e.g., Euclidean or any valid metric).

Let (Z_1, A) be a complete graph whose the nodes are the samples in Z_1. We define a path as being a sequence of distinct samples $\pi = \langle s_1, s_2, \ldots, s_k \rangle$, where $(s_i, s_{i+1}) \in A$ for $1 \leq i \leq k-1$. A path is said *trivial* if $\pi = \langle s_1 \rangle$. We assign to each path π a cost $f(\pi)$ given by a path-cost function f. A path π is said optimum if $f(\pi) \leq f(\pi')$ for any other path π', where π and π' end at a same sample s_k. We also denote by $\pi \cdot \langle s,t \rangle$ the concatenation of a path π with terminus at s and an arc (s,t). The OPF algorithm uses the path-cost function f_{max}, because of its theoretical properties for estimating optimum prototypes:

$$\begin{aligned} f_{max}(\langle s \rangle) &= \begin{cases} 0 & \text{if } s \in S, \\ +\infty & \text{otherwise} \end{cases} \\ f_{max}(\pi \cdot \langle s,t \rangle) &= \max\{f_{max}(\pi), d(s,t)\} \end{aligned} \tag{12}$$

We can observe that $f_{max}(\pi)$ computes the maximum distance between adjacent samples in π, when π is not a trivial path.

The OPF algorithm assigns one optimum path $P^*(s)$ from S to every sample $s \in Z_1$, forming an optimum path forest P (a function with no cycles which assigns to each $s \in Z_1 \backslash S$ its predecessor $P(s)$ in $P^*(s)$ or a marker nil when $s \in S$. Let $R(s) \in S$ be the root of $P^*(s)$ which can be reached from $P(s)$. The OPF algorithm computes for each $s \in Z_1$, the cost $C(s)$ of $P^*(s)$, the label $L(s) = \lambda(R(s))$, and the predecessor $P(s)$, as follows.

Algorithm 1 – OPF ALGORITHM

INPUT: A λ-labeled training set Z_1, prototypes $S \subset Z_1$ and the pair (v, d) for feature vector and distance computations.
OUTPUT: Optimum path forest P, cost map C and label map L.
AUXILIARY: Priority queue Q and cost variable cst.

1. *For each* $s \in Z_1 \backslash S$, *set* $C(s) \leftarrow +\infty$.
2. *For each* $s \in S$, *do*
3. $\quad$ $C(s) \leftarrow 0$, $P(s) \leftarrow nil$, $L(s) \leftarrow \lambda(s)$, *and insert* s *in* Q.
4. *While* Q *is not empty, do*
5. $\quad$ *Remove from* Q *a sample* s *such that* $C(s)$ *is minimum.*
6. $\quad$ *For each* $t \in Z_1$ *such that* $t \neq s$ *and* $C(t) > C(s)$, *do*
7. $\quad\quad$ *Compute* $cst \leftarrow \max\{C(s), d(s,t)\}$.
8. $\quad\quad$ *If* $cst < C(t)$, *then*
9. $\quad\quad\quad$ *If* $t \in Q$, *then remove* t *from* Q.
10. $\quad\quad\quad$ $P(t) \leftarrow s$, $L(t) \leftarrow L(s)$, $C(t) \leftarrow cst$, *and insert* t *in* Q.

Lines $1-3$ initialize maps and insert prototypes in Q. The main loop computes an optimum path from S to every sample s in a non-decreasing order of cost (Lines $4-10$). At each iteration, a path of minimum cost $C(s)$ is obtained in P when we remove its last node s from Q (Line 5). Lines $8-10$ evaluate if the path that reaches an adjacent node t through s is cheaper than the current path with terminus t and update the position of t in Q, $C(t)$, $L(t)$ and $P(t)$ accordingly. The label $L(s)$ may be different from $\lambda(s)$, leading to classification errors in Z_1. The training finds prototypes with none classification errors in Z_1. The OPF algorithm works with two phases: training and classification (test), as follows.

4.1 Training phase

We say that S^* is an optimum set of prototypes when Algorithm 1 propagates the labels $L(s) = \lambda(s)$ for every $s \in Z_1$. Set S^* can be found by exploiting the theoretical relation between *Minimum Spanning Tree* (MST) [9] and optimum path tree for f_{max}. The training essentially consists of finding S^* and an OPF classifier rooted at S^*.

By computing an MST in the complete graph (Z_1, A), we obtain a connected acyclic graph whose nodes are all samples in Z_1 and the arcs are undirected and weighted by the distance d between the adjacent sample feature vectors. This spanning tree is optimum in the sense that the sum of its arc weights is minimum as compared to any other

spanning tree in the complete graph. In the MST, every pair of samples is connected by a single path which is optimum according to f_{max}. That is, for any given sample $s \in Z_1$, it is possible to direct the arcs of the MST such that the result will be an optimum path tree P for f_{max} rooted at s. The optimum prototypes are the closest elements in the MST with different labels in Z_1. By removing the arcs between different classes, their adjacent samples become prototypes in S^* and Algorithm 1 can compute an optimum path forest with none classification errors in Z_1 without data overfitting [10].

4.2 Classification

For any sample $t \in Z_3$, the OPF consider all arcs connecting t with samples $s \in Z_1$, as though t were part of the graph. Considering all possible paths from S^* to t, we wish to find the optimum path $P^*(t)$ from S^* and label t with the class $\lambda(R(t))$ of its most strongly connected prototype $R(t) \in S^*$. This path can be identified incrementally, by evaluating the optimum cost $C(t)$ as

$$C(t) = \min\{\max\{C(s), d(s,t)\}\}, \ \forall s \in Z_1. \tag{13}$$

Let the node $s^* \in Z_1$ be the one that satisfies the above equation (i.e., the predecessor $P(t)$ in the optimum path $P^*(t)$). Given that $L(s^*) = \lambda(R(t))$, the classification simply assigns $L(s^*)$ as the class of t. An error occurs when $L(s^*) \neq \lambda(t)$.

4.3 Learning algorithm

The performance of the OPF classifier improves when the closest samples from different classes are included in Z_1, because the method finds prototypes that will work as sentinels in the frontier between classes. The learning algorithm replaces *irrelevant* samples of Z_1 by errors in Z_2, and secondly other samples of Z_2 are replaced by *irrelevant* samples of Z_1. In both cases, the algorithm never let a sample of Z_2 return to Z_1. The learning algorithm essentially tries to identify these prototypes from a few iterations of classification over Z_2.

In order to define irrelevant samples, the OPF algorithm can identify all samples in Z_1 that participated in the classification task of any node $t \in Z_2$. The OPF consider all right and wrong classifications in Z_2. When $t \in Z_2$ is correctly/incorrectly classified, we add one to the number of right/wrong classifications, of every sample $r \in Z_1$ in the optimum path $P^*(t)$ from $R(t) \in S^*$ to s^*. In that way, the learning algorithm outputs the final projected classifier, which can be now used to predict the labels of Z_3.

5 Experimental setup

5.1 Datasets

To evaluate the accuracy of our approach, we selected thirteen texture images obtained from the standard Brodatz database. Before being digitized, each of the 512×512 texture images was rotated at different degrees [11]. Figure 1 displays the non-rotated version of each of the texture images.

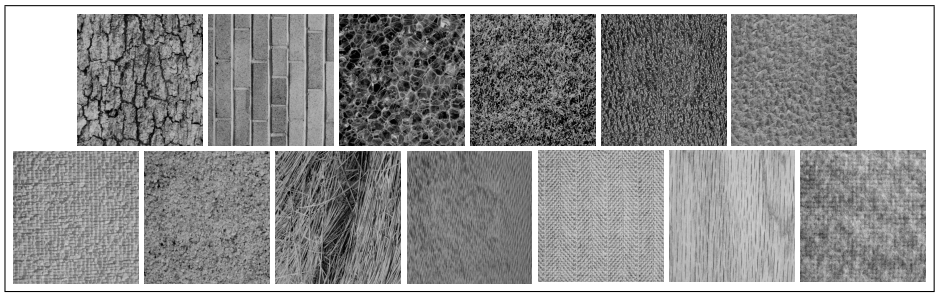

Fig. 1. Texture images from the Brodatz dataset used in our experiments. From left to right, and from top to bottom, they include: Bark, Brick, Bubbles, Grass, Leather, Pigskin, Raffia, Sand, Straw, Water, Weave, Wood, and Wool.

From this database, three different image datasets were generated: *non-distorted*, *rotated-set A*, and *rotated-set B*. The *non-distorted* image dataset was constructed just from the original input textures (i.e. texture patterns at 0 degrees). Each texture image was partitioned into sixteen 128×128 non-overlapping subimages. Thus, this dataset comprises 208 $(13{\times}16)$ different images. Furthermore, images belonging to this dataset will be used in the learning stage of our classifier. The second image dataset is referred to as *rotated image dataset A*, and was generated by selecting the four 128×128 innermost subimages from texture images at $0, 30, 60,$ and 120 degrees. A total number of 208 images were generated ($13 \times 4 \times 4$). Finally, in the *rotated image dataset B*, we selected the four 128×128 innermost subimages of the rotated image textures (512×512) at $0, 30, 60, 90, 120, 150$ and 200 degrees. This led to 364 $(13 \times 4 \times 4)$ dataset images. *Rotated image dataset A*, as well as *rotated image dataset B* will be used for recognition purposes.

5.2 Classification evaluation

In our experiments, the accuracy was measured by taking into account the possibility of having classes with different cardinalities. Let $N(i)$, $i = 1, 2, \ldots, c$, be the number of samples in each class i and $N = N(1) \cup N(2) \ldots \cup N(c)$ be the whole dataset. We define the partial errors $e_{i,1}$ and $e_{i,2}$ as follows:

$$e_{i,1} = \frac{FP(i)}{|N| - |N(i)|} \text{ and } e_{i,2} = \frac{FN(i)}{|N(i)|},\ i = 1, \ldots, c. \tag{14}$$

where $FP(i)$ and $FN(i)$ denote both false positives and false negatives, respectively. That is, $FP(i)$ represents the number of samples of other classes that were classified as being from the class i. In addition, $FN(i)$ represents the number of samples in class i that were misclassified. The errors $e_{i,1}$ and $e_{i,2}$ are then used to define the accumulated partial error of class i as:

$$E(i) = e_{i,1} + e_{i,2}. \tag{15}$$

Finally, the resulting classification accuracy $\mathcal{L}$ is found:

$$\mathcal{L} = \frac{2c - \sum_{i=1}^{c} E(i)}{2c} = 1 - \frac{\sum_{i=1}^{c} E(i)}{2c}. \quad (16)$$

6 Experimental results

To demonstrate the discriminating power of our proposed method for recognizing rotated texture patterns, we conducted two series of experiments. In the first series of experiments (Subsection 6.1), we evaluated the effectiveness of the proposed rotation-invariant representation against two other approaches: the conventional Pyramid Decomposition [12] and with a recent proposal based on Gabor Wavelets [13]. Furthermore, the second series of experiments (Subsection 6.2), were used to evaluate the recognition accuracy of the novel OPF multi-class classifier. Note that, both series of experiments were conducted using *rotated-sets A* and *B*. The *rotated-set A* was used to analyze the recognition accuracy of our method under the presence of few rotated versions of texture patterns. In addition, experiments conducted in the *rotated-set B* consider several rotated versions of different texture patterns.

In both series of experiments, we used Steerable Pyramids having different decomposition levels ($S = 2, 3, 4$) at several orientations ($K = 4, 5, 6, 7, 8$). Our experiments agree with [14] in that, the most relevant textural information in images is contained in the first two levels of decomposition, since little recognition improvement is achieved by varying the number of scales during image decomposition. Therefore, we focus our discussions on image decompositions having ($S = 2, 3$) scales. The dimensionality of the feature vectors depends on the number of scales (S) and on the number of orientations (O) considered during image decomposition and it is computed as follows: $2 \times O \times S$. Furthermore, an important motivation in our study was to use small size feature vectors, in order: (1) to show that the recognition accuracy of our approach is not compromised, and (2) to facilitate texture recognition applications where data storage capacity is a limitation.

6.1 Effectiveness of the rotation invariance representation

To analyze the texture characterization capabilities of our method against the conventional Pyramid Decomposition [12] and the Gabor Wavelets [13], we used Gaussian-kernel Support Vector Machines (SVMs) as texture classification mechanisms. Note that, the SVM parameters were optimized by using the cross-validation method [1].

Figure 2 compares the recognition accuracy obtained by those three methods in the *rotated-set A*, whereas Figure 3 depicts the recognition accuracy obtained in the *rotated-set B*. From both figures, it can be seen that our image descriptor outperforms the other two approaches, regardless of the number of scales or orientations used for extracting the feature vectors.

In the case of the *rotated-set A*, the higher classification accuracies achieved by our method were obtained by using 7 orientations, which corresponds to image rotations in

[1] We used the well known LIBSVM package [15]

steps of $25.71°$. Those accuracies are respectively 100% and 97.31% for two and three decomposition levels ($s = 2, 3$). The corresponing classification accuracies obtained by the Gabor Wavelets are: 90.36% and 93.90% ($s = 2, 3; o = 7$), whereas for the conventional Steerable Pyramid those accuracies are: 89.67% and 90.36%. Furthermore, the accuracies using $o = 6, 7, 8$ orientations are very close to each other. Therefore, $s = 2$ and $o = 6$ are the most appropiate parameter combinations for our rotation-invariant image descriptor, and at the same time, low dimensionality feature vectors are obtained.

In the case of the *rotated-set B*, the higher classification accuracies achieved by our descriptor, were again obtained by using 7 orientations. Classification rates of 95.86% and 95.73% correspond respectively to feature vectors with $s = 2, 3$ scales and $o = 7$ orientations. Further, it is found that both Gabor Wavelets and Conventional Steerable Pyramid Decomposition present lower classification rates, being respectively 91.05% and 95.35% for the first method, and 84.22%, 84.23% for the second one. As in the results obtained in *rotated-set A*, we can notice that the achieved classification accuracies are very close to each other, when using $o = 6, 7$ or $o = 8$ orientations. From these results, we can reinforce that the most appropriate parameter settings for our descriptor are $s = 2$ scales and $o = 6$ orientations.

Furthermore, from the bar graphs shown in Figures 2 and 3, the best performance obtained by the rotation-invariant Gabor method is as good as our descriptor. However, this rate is obtained at $s = 3$ scales, whereas the proposed descriptor achieves the same performance using only $s = 2$ scales. In this sense, an important advantage of our method is its high performance rate at low size feature vectors. It is worth to mention that in our experiments the OPF classifier was at least 10 times faster than the SVM classifier[2].

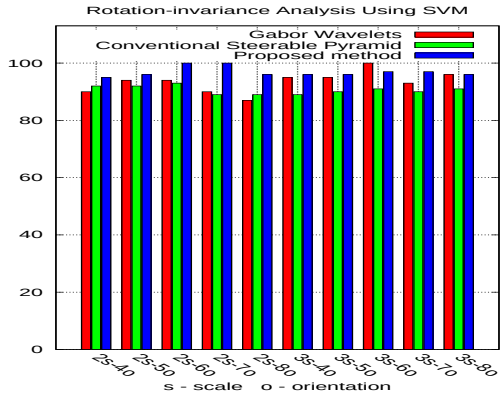

Fig. 2. Classification accuracy comparison using the SVM classifier obtained in *rotated-set A* using ($S = 2, 3$) scales with ($K = 4, 5, 6, 7, 8$) orientations for Gabor Wavelets, conventional Steerable Pyramid decomposition and our method.

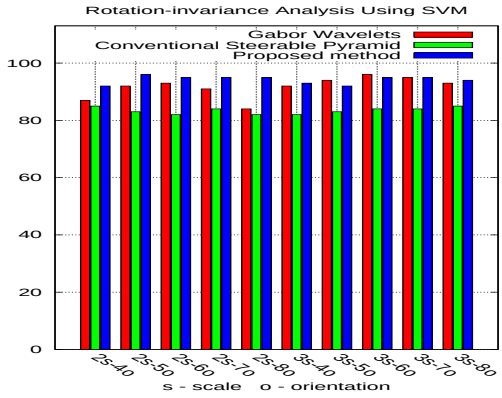

Fig. 3. Classification accuracy comparison using the SVM classifier obtained in *rotated-set B* using ($S = 2, 3$) scales with ($K = 4, 5, 6, 7, 8$) orientations for Gabor Wavelets, conventional Steerable Pyramid decomposition and our method.

[2] Due to the lack of space, we could not present a detailed study of the execution time

6.2 Effectiveness of the multiclass recognition method

In the previous subsection, we showed that our proposed rotation-invariant image descriptor outperforms the other two methods. Therefore, our objective now is to show the recognition improvement of our classifier over the SVM approach. It can be seen from Figure 4, that for almost all feature extraction configurations, the recognition rates of the OPF classifier are higher than those of the SVM classifier. However, the latter method presents the same recognition rates as the OPF classifier when using $s = 2$ scales and $o = 6, 7$ orientations. In the case of the image *rotated-set B*, our classifier yields better recognition rates for all feature extraction configurations (See Figure 5). By considering that it was found, that the most appropriate parameter settings for our descriptor are $s = 2$ scales and $o = 6$ orientations, it is worth to mention, that by using this configuration, the recognition accuracy obtained by the OPF classifier is 98.49% in comparison with the corresponding accuracy of 95.48% obtained by the SVM classifier.

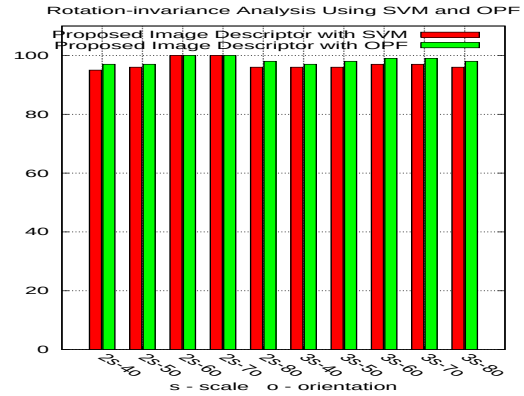

Fig. 4. Recognition accuracy comparison of the OPF and SVM classifiers in *rotated-set A* using ($S = 2, 3$) scales with ($K = 4, 5, 6, 7, 8$) orientations for our rotation-invariant image descriptor.

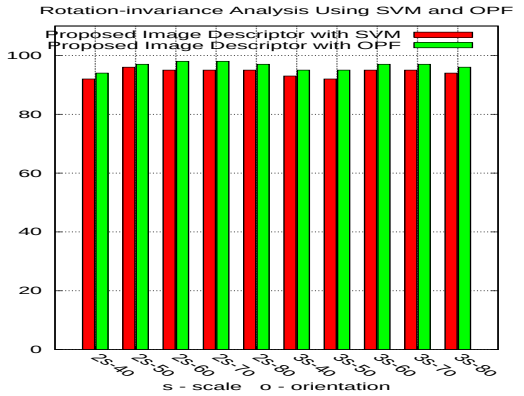

Fig. 5. Recognition accuracy comparison of the OPF and SVM classifiers in *rotated-set B* using ($S = 2, 3$) scales with ($K = 4, 5, 6, 7, 8$) orientations for our rotation-invariant image descriptor.

6.3 Results Summarization

A summary of our experimental results is provided in Tables 1, and 2. Table 1 compares for each dataset, the mean recognition rates obtained by the three texture image descriptors using different scales ($s = 2, 3$) and different orientations ($o = 4, 5, 6, 7, 8$). In this set of experiments, we used Gaussian-kernel Support Vector Machines (SVMs) as texture classification mechanisms. From our results, it can be noticed that our texture image descriptor performs better regardless of the dataset used, or the image decomposition parameters considered during feature extraction (number of scales and orientations). Furthermore, the second series of experiments are summarized in Table 2. As it can be

seen, the OPF classifier improves the recognition accuracies obtained by the SVM classifier in all of our experiments. By considering the summarized results, it can be shown that our proposed recognition system performs better than the previously mentioned approaches, which represent state-of-the-art methods.

Rotated Image Dataset	Feature vectors with different scales (s) and orientations (o)	Proposed Image Descriptor	Conventional Steerable Pyramid Decomposition	Gabor Wavelets
A	(s=2;o=4,5,6,7,8)	**98.89%**	93.19%	93.19%
A	(s=3;o=4,5,6,7,8)	**98.61%**	92.36%	97.92%
B	(s=2;o=4,5,6,7,8)	**97.35%**	85.30%	91.29%
B	(s=3;o=4,5,6,7,8)	**96.74%**	85.76%	96.67%

Table 1. Mean recognition rates for the three different texture image descriptors using Gaussian-kernel Support Vector Machines as classifiers.

Rotated Image Dataset	Feature vectors with different scales (s) and orientations (o)	Proposed Image Descriptor using OPF	Proposed Image Descriptor using SVM
A	(s=2;o=4,5,6,7,8)	**98.89%**	95.89%
A	(s=3;o=4,5,6,7,8)	**98.61%**	97.99%
B	(s=2;o=4,5,6,7,8)	**97.35%**	92.30%
B	(s=3;o=4,5,6,7,8)	**96.74%**	96.70%

Table 2. Mean recognition rates for the proposed rotation-invariant texture image descriptor using both OPF and SVM classifiers.

7 Conclusions

In this paper a new novel texture classification system was proposed. Its main features include: (1) a new rotation-invariant image descriptor, and (2) a novel multi-class recognition method based on Optimum Path Forest. The proposed image descriptor exploits the discriminability properties of the Steerable Pyramid Decomposition for texture characterization. To obtain rotation-invariance, the dominant orientation of the input textures is found, so that feature elements are aligned according to this value. By doing this, a more reliable feature extraction process can be performed, since corresponding feature elements of distinct feature vectors, coincide with images at same orientations. In addition, our system adopted a novel approach for pattern classification based on Optimum Path Forest, which finds prototypes with none zero classification errors in the training sets and learns from errors in evaluation sets. By combining the discriminating power of our image descriptor and classifier, our system uses small size feature vectors to characterize texture images without compromising overall classification rates, being ideally for real-time applications.

Furthermore, we have demonstrated state-of-the-art results on two image datasets derived from the standard Brodatz database. For the first image dataset, our method obtained a mean classification rate of 98.89% in comparision with a mean accuracy of 93.19% obtained by both conventional Steerable Pyramid decomposition [12] and Gabor Wavelets [13]. For the second image dataset, our method achieved a mean classification of 97.35%, whereas the other two methods obtained respectively classification rates of 85.30% and 91.29%.

On the other side, the OPF multi-class classifier outperformed the SVM in the two datasets. It is a new promising graph tool for pattern recognition, which differs from traditional approaches, in that, it does not use the idea of feature space space geometry, therefore, better results in overlapped databases are achieved. Future work will include extending this method for scale-invariant texture recognition.

This work is partially supported by Microsoft Escience and Tablet PC Technology and Higher Education projects, CNPq (Proc. 134990/2005-6, 477039/2006-5 and 311309/2006-2), FAPESP, CAPES, and FAEPEX.

References

1. Haykin, S.: Neural networks: a comprehensive foundation. Prentice Hall (1994)
2. Boser, B., Guyon, I., Vapnik, V.: A training algorithm for optimal margin classifiers. In: Proc. 5th Workshop on Computational Learning Theory. (1992) 144–152
3. Papa, J.P., Falcão, A.X., Miranda, P.A.V., Suzuki, C.T.N., Mascarenhas, N.D.A.: A new pattern classifier based on optimum path forest. Technical Report IC-07-13, Institute of Computing, State University of Campinas (2007) Technical report available at http://www.dcc.unicamp.br/ic-tr-ftp/2007/07-13.ps.gz.
4. Freeman, W.T., Adelson, E.H.: The design and use of steerable filters. IEEE Trans. Pattern Anal. Mach. Intell. **13** (1991) 891–906
5. Portilla, J., Simoncelli, E.P.: A parametric texture model based on joint statistics of complex wavelet coefficients. International Journal of Computer Vision **40** (2000) 49–70
6. Bimbo, A.D.: Visual information retrieval. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1999)
7. Zhang, D., Wong, A., Indrawan, M., Lu, G.: Content based image retrieval using gabor texture features. In: PCM'00: Proc. of 1st IEEE Pacific-Rim Conf. on Multimedia. (2000) 392–395
8. Falcão, A., Stolfi, J., Lotufo, R.: The image foresting transform: theory, algorithms, and applications. IEEE Trans. Pattern Anal. Mach. Intell. **26** (2004) 19–29
9. Cormen, T., Leiserson, C., Rivest, R.: Introduction to Algorithms. MIT (1990)
10. Cousty, J., Bertrand, G., Najman, L., M.Couprie: Watersheds, minimum spanning forests, and the drop of water principle. (2007) École Supérieure d'Ingénieurs.
11. University of Southern California, S., Institute, I.P.: (Rotated texture database) http://sipi.usc.edu/services/database/Database.html (Accessed on 1 March 2007).
12. Simoncelli, E.P., Freeman, W.T.: The steerable pyramid: A flexible architecture for multi-scale derivative computation. Proceedings of IEEE ICIP **13** (1995) 891–906
13. Arivazhagan, S., Ganesan, L., Priyal, S.P.: Texture classification using gabor wavelets based rotation invariant features. Pattern Recognition Letters **27** (2006) 1976–1982
14. Do, M.N., Vetterli, M.: Wavelet-based texture retrieval using generalized gaussian density and kullback-leibler distance. IEEE Transactions on Image Processing **11** (2002) 146–158
15. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. (2001) Software available at urlhttp://www.csie.ntu.edu.tw/ cjlin/libsvm.